said selected scripts proceed to a second of said processing modules for processing of a next instruction within said instances of said selected scripts.

## REMARKS

Firstly, Applicants and their representatives thank Examiner Susan Lao and SPE Alvin Oberley for the courtesy of an in-person interview held on July 24, 2001 and telephonic interviews subsequent thereto. The substance of those interviews is contained within the remarks that follow.

In the present Office Action claims 1-46 and 60-63 were examined. Claims 1-46 and 60-63 were rejected, and no claims were allowed. The Office Action makes the rejections final.

By this Amendment and Response, claims 1, 4-14, 16, 29-34, 37, 38, 41, 42, 60, 62 and 63 are proposed to be amended, no claims are proposed to be cancelled or added. Accordingly, claims 1-46 and 60-63 are still pending in the application and, as now written, are believed to be in a condition for allowance.

Support for the proposed amendments to independent claims 1, 33, 34, 60 and 63 may be found in the application and drawings as filed, and at least at page 37, lines 10-20 and page 26, line 24 to page 27, line 13. Thus, no new matter was added.

Reconsideration and allowance in view of the proposed amendments and remarks to follow is respectfully requested.

Prior Art Rejections:

Claims 1-4, 20-27, 34-35, 41-44, 46 and 63 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Krishnamurthy et al. (a technical journal article) in view of Mahajan (U.S. Patent No. 5,404,528). Additionally, claims 5-19, 28-33, 36-40, 45 and 60-62 stand rejected under

35 U.S.C. 103(a) as being unpatentable over Krishnamurthy et al. in view of Mahajan as applied to claim 1, and further in view of Waclawsky et al. (U.S. Patent No. 5,493,689). These rejections are respectfully disagreed with and are traversed below.

The cited documents are merely seen to describe a static execution and/or interpretation of a script by one processing module. Applicants submit that the present invention teaches and now claims, inter alia, an evaluation of dynamic information, comprising statuses of processing modules and processed data provided by the processing modules, for selection of which of multiple processing modules process instructions within a script.

As noted above, clarifying amendments are proposed to even further distinguish the independent claims from the cited documents. In view of the foregoing, Applicants submit that independent claims, and the claims that depend therefrom, are patentable over the cited documents.

Accordingly, the Examiner is respectfully requested to enter the proposed amendments, to reconsider and remove the rejections of all of the pending claims and to allow the application as now presented. If a notice of allowance cannot be issued, it is respectfully requested that the undersigned attorney of record be contacted to resolve any outstanding issues.

Respectfully submitted

DATE: August 10, 2001

Michael K. Kinney, Reg. No. 42,740
Attorney for Applicants

WIGGIN & DANA, LLP
One Century Tower
New Haven, Connecticut 06508
**Tel. No. 203-498-4411**

66SH01!.DOC\12217\1\288737.01

## MARKED-UP VERSION OF CLAIMS

1. (Thrice Amended) A data processing [system stored on a computer-readable medium] system, comprising:

a plurality of event modules each including code that generates an event data signal representative of a particular event;

a plurality of scripts each having a plurality of instructions [that provide results];

a plurality of processing modules distributed over said data processing system each including code that provides processed data [to said plurality of scripts]; and

a task module, selectively communicating with each of said plurality of event modules and said plurality of distributed processing modules, said task module including code for [executing a selected] selecting and instantiating one of said plurality of scripts that corresponds to said event data signal [, for executing said plurality of instructions within said selected script,] and for [selectively invoking at least] executing said instance of said selected script such that said instance of said selected script proceeds to [one] a first of said plurality of distributed processing modules [in accordance with information regarding statuses of said plurality of processing modules and results generated by previously executed] for processing a current one of said plurality of instructions;

wherein dynamic information comprises statuses of said distributed processing modules and said processed data, and wherein during execution of said instance of said selected script [, said selected script incorporates said processed data to modify execution of next instructions] said task module provides said dynamic information to said instance of said selected script for incorporation therein, and upon completion of said currently executing instruction, said task module evaluates said incorporated dynamic information and selectively executes, based upon said

incorporated dynamic information, said instance of said selected script such that said instance of said selected script proceeds to a second of said distributed processing modules for processing a next instruction within said instance of said selected script.

2. The system as claimed in claim 1 wherein said task module executes two or more of said plurality of scripts substantially simultaneously.

3. The system as claimed in claim 2 further comprising:

a converter module, in communication with said task module, including code that maps said event data signal to at least one of said two or more of said plurality of scripts upon reception of said event data signal by said task module.

4. (Twice Amended) The system as claimed in claim 1 wherein said plurality of distributed processing modules provide event data signals, representative of particular events, to said task module.

5. (Twice Amended) The system as claimed in claim 1 further comprising:

a status monitoring module, in communication with said task module, including code that provides said status information to said task module including operating conditions of said plurality of distributed processing modules.

6. (Twice Amended) The system as claimed in claim 5 wherein said status monitoring module is in direct communication with said plurality of distributed processing modules.

7. (Amended) The system as claimed in claim 5 wherein during said execution of said instance of said selected script, said status monitoring module stores data associated with said instance of said selected script in an associated memory.

8. (Twice Amended) The system as claimed in claim 1 further comprising:

a load balancing module, in communication with said task module, including code that dynamically selects available ones of said plurality of distributed processing modules to perform processing.

9. (Twice Amended) The system as claimed in claim 8 wherein said load balancing module is in direct communication with said plurality of distributed processing modules.

10. (Twice Amended) The system as claimed in claim 1 wherein said task module interfaces with said plurality of distributed processing modules for bi-directionally and substantially simultaneously transmitting data between said plurality of distributed processing modules and said task module.

11. (Twice Amended) The system as claimed in claim 1 further comprising:

a resource management module, in communication with said task module, including code [that dynamically assigns processing functions to said plurality of processing modules] for monitoring event data signals generated by said plurality of event modules and not processed by said task module and a number of said plurality of distributed processing modules available for

performing particular data processing functionality, and for converting data processing functionality of said plurality of distributed processing modules in response to dynamic information regarding said monitored event data signals and said number of available distributed processing modules to maximize a number of said distributed processing modules processing said event data signals.

12. (Twice Amended) The system as claimed in claim 11 wherein said resource management module is in direct communication with said plurality of distributed processing modules.

13. (Twice Amended) The system as claimed in claim 1 further comprising:

a plurality of initiator modules including code that provides a communication interface between an associated one of said plurality of distributed processing modules and said task module.

14. (Twice Amended) The system as claimed in claim 13 wherein each of said plurality of initiator modules communicates with said associated one of said plurality of distributed processing modules regardless of native applications contained on said associated one of said plurality of distributed processing modules.

15. The system as claimed in claim 13 further comprising:

a protocol disposed between each of said plurality of initiator modules and said task module for providing a communication interface therebetween.

16. (Twice Amended) The system as claimed in claim 13 further comprising:

a protocol disposed between each of said plurality of initiator modules and said associated one of said plurality of distributed processing modules for providing a communication interface therebetween.

17. The system as claimed in claim 1 further comprising:

a plurality of client modules including code that provides a communication interface between an associated one of said plurality of event modules and said task module.

18. The system as described in claim 17 further comprising:

a protocol disposed between said task module and each of said plurality of client modules for providing a communication interface therebetween.

19. The system as claimed in claim 17 further comprising:

a protocol disposed between each of said plurality of client modules and said associated one of said plurality of event modules for providing a communication interface therebetween.

20. The system as claimed in claim 1 wherein each of said plurality of scripts is preprogrammed to iteratively update its contents.

21. The system as claimed in claim 1 further comprising:

a storage module, in communication with said task module, for providing storage for said system.

22. The system as claimed in claim 21 wherein said storage module comprises a computer-readable medium.

23. The system as claimed in claim 22 wherein said computer readable medium comprises a persistent memory.

24. The system as claimed in claim 21 further comprising:

a script building module, in communication with said storage module, including code that creates said plurality of scripts.

25. The system as claimed in claim 24 wherein said script building module includes a standard language interface.

26. The system as claimed in claim 24 wherein said script building module includes a graphical user interface.

27. The system as claimed in claim 24 wherein said script building module dynamically updates and modifies said plurality of scripts.

28. The system as claimed in claim 1 further comprising:

a protocol for providing a communication interface between said task module and each of said plurality of event modules.

29. (Twice Amended) The system as claimed in claim 1 further comprising:

a protocol for providing a communication interface between said task module and each of said plurality of distributed processing modules.

30. (Thrice Amended) The system as claimed in claim 1 further comprising:

a responder module, in communication with said task module, including code that transmits response data, resulting from said execution, from said task module in a particular format to said plurality of distributed processing modules or in a particular format to said plurality of event modules.

31. (Twice Amended) The system as claimed in claim 1 further comprising:

an administrative module, in communication with said task module, including code that receives and presents data that relates to said plurality of distributed processing modules.

32. (Twice Amended) The system as claimed in claim 1 further comprising:

a plurality of application peripherals in communication with an associated one of said plurality of distributed processing modules or an associated one of said plurality of event modules.

33. (Thrice Amended) A data processing [system stored on a computer readable-medium] system, comprising:

a plurality of event modules each including code that generates an event data signal representative of a particular event;

a plurality of scripts each having a plurality of instructions;

a plurality of processing modules distributed over said data processing system each including code [that provides said plurality of scripts with] for performing data processing functionality to provide processed data;

a task module, selectively communicating with each of said plurality of event modules and said plurality of distributed processing modules, said task module including code for [executing a selected] selecting and instantiating one of said plurality of scripts that correspond to said event data signal and, during execution of said instance of said selected script, for [executing said plurality of instructions within said selected script, and for selectively invoking at least one of said plurality of process modules in accordance with] providing dynamic information [regarding a status] comprising statuses of said plurality of distributed processing modules and [results generated by previously executed instructions] said processed data to said instance of said selected script for incorporation therein and, for selectively executing, based on said incorporated dynamic information, said instance of said selected script such that said instance of said selected script proceeds to a first and to at least a second of said distributed processing modules for processing instructions within said instance of said selected script; and

[wherein during execution of said selected script said selected script incorporates said processed data to modify execution of next instructions of said selected script;]

a resource management module [in direct communication with] communicating with each of said plurality of event modules, said task module and said plurality of distributed processing modules, said resource management module including code [that dynamically assigns processing

functions to said plurality of processing modules; and] for monitoring event data signals generated by said plurality of event modules and not processed by said task module and a number of said plurality of distributed processing modules available for performing particular data processing functionality, and for converting data processing functionality of said plurality of distributed processing modules in response to dynamic information regarding said monitored event data signals and said number of available distributed processing modules to maximize a number of said distributed processing modules processing said event data signals

[an administrative module in direct communication with said task module, said plurality of event modules, said plurality of processing modules and said resource management module, including code that receives and presents data relating to said plurality of processing modules].

34. (Thrice Amended) A method of data processing comprising the steps of:

generating at least one event data signal at one or more peripheral modules;

mapping said at least one event data signal to a selected script chosen from one or more scripts, each said one or more scripts having one or more instructions [for performing data gathering steps];

instantiating said selected script; and

executing [said one or more instructions within said selected script; and

invoking], by a task module, said instance of said selected script such that said instance of said selected script proceeds to a first of [in accordance with status information, one or more] a plurality of processing modules [to process data required by] for processing a current one of said one or more instructions of said instance of [one or more] said selected script;

wherein dynamic information comprises statuses of said plurality of processing modules and processed data provided by said plurality of processing modules, and wherein during execution of said [one or more instructions,] instance of said selected script [dynamically incorporates data processed by said one or more processing modules to modify execution of next ones of said one or more instructions] said task module provides said dynamic information to said instance of said selected script for incorporation therein, and upon completion of said currently executing instruction, said task module evaluates said incorporated dynamic information and selectively executes, based upon said incorporated dynamic information, said instance of said selected script such that said instance of said selected script proceeds to a second of said plurality of processing modules for processing a next instruction within said instance of said selected script.

35. The method as claimed in claim 34 wherein said one or more peripheral modules and said task module communicate via a communication interface.

36. The method as claimed in claim 34 further comprising the step of:

dynamically managing operating functions of said one or more peripheral modules.

37. (Amended) The method as claimed in claim 34 further comprising the steps of:

producing response data signals as a result of said [executing step] execution of said instance of said selected script; and

transmitting said response data signals from said task module to selected ones of said one or more peripheral modules.

38. (Amended) The method as claimed in claim 37 further comprising the step of:

translating said response data signals transmitted from said task module into a format that said selected ones of said one or more peripheral modules recognize.

39. The method as claimed in claim 38 further comprising the step of:

storing said event data signals, said one or more scripts and said response data signals in a storage medium that is in communication with said task module.

40. The method as claimed in claim 39 wherein said storage medium is persistent.

41. (Amended) The method as claimed in claim 34 further comprising the step of:

accessing a protocol to interface between said task module and selected ones of said one or more peripheral modules.

42. (Twice Amended) The method as claimed in claim 34 further comprising the step of:

providing communication between said task module and each of said [one or more peripheral] plurality of processing modules such that said [task module invokes] instance of said selected script proceeds to only ones of said [one or more peripheral] plurality of processing modules available for performing processing operations.

43. The method as claimed in claim 34 wherein said executing step includes the step of:

interfacing with a plurality of said one or more peripheral modules substantially simultaneously.

44. The method as claimed in claim 34 wherein said executing step executes a plurality of said one or more scripts substantially simultaneously.

45. The method as claimed in claim 34 wherein said execution of said one or more instructions dynamically incorporates data gathered in previously executed instructions.

46. The method as claimed in claim 34 further comprising the step of:

providing results of said executing step to an administrative module for presenting information relating to said one or more peripheral modules.

60. (Amended) In a data processing system, a method for responding to event data, comprising:

receiving event data from a requesting one of a plurality of event modules;

mapping the event data to a selected one of a plurality of scripts, the plurality of scripts including instructions for responding to event data;

instantiating said selected script;

executing, by a task module, the [instructions within the mapped script to generate results] instance of the selected script such that the instance of the selected script proceeds to a first of a plurality of processing modules for processing of a current one of the instructions of the instance of the selected script;

wherein dynamic information comprises statuses of the plurality of processing modules and processed data provided by the processing modules, and wherein during the execution of [at least one instruction, selecting and invoking one of a plurality of processing modules available for providing processed data to the at least one instruction, the selecting and invoking in accordance with information regarding a status of the processing modules and results generated by previously executed instructions, and modifying execution of next instructions within the mapped script in accordance with the processed data generated by the selected and invoked processing module] the instance of the selected script the task module provides the dynamic information to the instance of the selected script for incorporation therein, and upon completion of the currently executing instruction the task module evaluates the incorporated dynamic information and selectively executes, based on the incorporated information, the instance of the selected script such that the instance of the selected script proceeds to a second of the plurality of processing modules for processing a next instruction within the instance of the selected script;

building a response profile including [the generated] results generated during execution of the instance of the selected script; and

wherein when the instructions within the [mapped] instance of the selected script are completed, transmitting the response profile to the requesting one of the plurality of event modules.

61. The method as claimed in claim 60 wherein the generated results include event data.

62. (Amended) The method as claimed in claim 60, comprising:

tracing execution of the instructions within the [mapped] instance of the selected script and processing of the [invoked] processing modules; and

wherein when a processing module fails, continuing execution of the [mapped] instance of the selected script and the processing of the [invoked] processing modules from a last traced instruction.

63. (Amended) A data processing [system stored on a computer-readable medium] system, comprising:

a plurality of event modules each including code that generates a first event data signal representative of a first event;

a plurality of scripts each having a plurality of instructions [that provide results];

a plurality of processing modules each including code that provides processed data, a subset of said plurality of processing modules having code that selectively generates a second event data signal representative of a second event; and

a task module, selectively communicating with each of said plurality of event modules and said [subset of said] plurality of processing modules, said task module including code for [executing selected] selecting and instantiating ones of said plurality of scripts that corresponds to said first and second event data signals, [for executing said plurality of instructions within said selected scripts,] and for [selectively invoking one of said plurality of processing modules in accordance with information regarding statuses of said plurality of processing modules and results generated by previously executed instructions] executing said instances of said selected scripts such that said instances of said selected scripts proceed to a first of said plurality of processing modules for processing a current one of said plurality of instructions within each of said instances;

wherein dynamic information comprises statuses of said plurality of processing modules and said processed data, and wherein during execution of said instances of said selected scripts, [each of said selected scripts incorporates said processed data to modify execution of next instructions of said selected scripts] said task module provides said dynamic information to said instances of said selected scripts for incorporation therein, and upon completion of said currently executing instructions, said task module evaluates said incorporated dynamic information and selectively executes, based on said incorporated dynamic information, said instances of said selected scripts such that said instances of said selected scripts proceed to a second of said processing modules for processing of a next instruction within said instances of said selected scripts.